


```

xxy=a.createxyvsy('new') # Copies content of 'default'

To Modify an existing Xyvsy use:
xxy=a.getxyvsy('AMIP_psl')

xxy.list() # Will list all the Xyvsy at
xxy.projection='linear' # Can only be 'linear'
lon30={-180:'180W',-150:'150W',0:'Eq'}
xxy.xticlabels1=lon30
xxy.xticlabels2=lon30
xxy.xticlabels(lon30, lon30) # Will set them both
xxy.xmtics1=''
xxy.xmtics2=''
xxy.xmtics(lon30, lon30) # Will set them both
xxy.yticlabels1=lat10
xxy.yticlabels2=lat10
xxy.yticlabels(lat10, lat10) # Will set them both
xxy.ymtics1=''
xxy.ymtics2=''
xxy.ymtics(lat10, lat10) # Will set them both
xxy.datawc_y1=-90.0
xxy.datawc_y2=90.0
xxy.datawc_x1=-180.0
xxy.datawc_x2=180.0
xxy.datawc(-90, 90, -180, 180) # Will set them all
xxy.yaxisconvert='linear'

Specify the Xyvsy line type:
xxy.line=0 # same as xxy.line = 'solid'
xxy.line=1 # same as xxy.line = 'dash'
xxy.line=2 # same as xxy.line = 'dot'
xxy.line=3 # same as xxy.line = 'dash-d'
xxy.line=4 # same as xxy.line = 'long-d'

Specify the Xyvsy line color:
xxy.linecolor=16 # color range: 16 to 230, default color is bl
xxy.linewidth=16 # width range: 1 to 100, default color is bl

Specify the Xyvsy marker type:
xxy.marker=1 # Same as xxy.marker='dot'
xxy.marker=2 # Same as xxy.marker='plus'
xxy.marker=3 # Same as xxy.marker='star'
xxy.marker=4 # Same as xxy.marker='circle'
xxy.marker=5 # Same as xxy.marker='cross'
xxy.marker=6 # Same as xxy.marker='diamond'
xxy.marker=7 # Same as xxy.marker='triang
xxy.marker=8 # Same as xxy.marker='triang
xxy.marker=9 # Same as xxy.marker='triang
xxy.marker=10 # Same as xxy.marker='triang
xxy.marker=11 # Same as xxy.marker='square'
xxy.marker=12 # Same as xxy.marker='diamond'
xxy.marker=13 # Same as xxy.marker='triang

```

```

xyy.marker=14           # Same as xyy.marker='triang
xyy.marker=15           # Same as xyy.marker='triang
xyy.marker=16           # Same as xyy.marker='triang
xyy.marker=17           # Same as xyy.marker='square
xyy.marker=None         # Draw no markers

```

There are four possibilities for setting the marker color index

```

xyy.markercolors=22     # Same as below
xyy.markercolors=(22)   # Same as below
xyy.markercolors=([22]) # Will set the markers to a
                        #         color index
xyy.markercolors=None   # Color index defaults to Bl

```

To set the Xyvsy Marker size:

```

xyy.markersize=5
xyy.markersize=55
xyy.markersize=100
xyy.markersize=300
xyy.markersize=None

```

Methods defined here:

```
__init__(self, parent, GXy_name=None, GXy_name_src='default', createGXy=0)
```

```
datawv(self, dsp1=1e+20, dsp2=1e+20, dsp3=1e+20, dsp4=1e+20)
```

```
list(self)
```

```
rename = renameGXy(self, old_name, new_name)
```

```

#####
#
# Function:      renameGXy
#
# Description of Function:
#     Private function that renames the name of an existing
#     graphics method.
#
#
# Example of Use:
#     renameGXy(old_name, new_name)
#         where: old_name is the current name of Xyvsy g
#                new_name is the new name for the Xyvsy
#
#####

```

```
script(self, script_filename=None, mode=None)
```

```
Function:      script                                     # Calls _vcs.s
```

```
Description of Function:
```

```
Saves out a Xyvsy graphics method in Python or VCS scri
designated file.
```

Example of Use:

```
script(scriptfile_name, mode)
```

where: scriptfile_name is the output name of the
mode is either "w" for replace or "a" for

Note: If the the filename has a ".py" at the end
Python script. If the filename has a ".scr"
produce a VCS script. If neither extension
default a Python script will be produced.

```
a=vcs.init()  
Xy=a.createboxfill('temp')  
Xy.script('filename.py')           # Append to a Python file  
Xy.script('filename.scr')         # Append to a VCS file "f  
Xy.script('filename','w')
```

xmtics(self, xmt1=",", xmt2=")

xticlabels(self, xtl1=",", xtl2=")

ymtics(self, ymt1=",", ymt2=")

yticlabels(self, ytl1=",", ytl2=")

Properties defined here:

datawc_calendar

```
get">get = _getcalendar(self)  
set">set = _setcalendar(self, value)
```

datawc_timeunits

```
get">get = _gettimeunits(self)  
set">set = _settimeunits(self, value)
```

datawc_x1

```
get">get = _getdatawc_x1(self)  
set">set = _setdatawc_x1(self, value)
```

datawc_x2

```
get">get = _getdatawc_x2(self)  
set">set = _setdatawc_x2(self, value)
```

datawc_y1

```
get">get = _getdatawc_y1(self)  
set">set = _setdatawc_y1(self, value)
```

datawc_y2

```
get">get = _getdatawc_y2(self)  
set">set = _setdatawc_y2(self, value)
```

line

```
get">get = _getline(self)
set">set = _setline(self, value)
```

linecolor

```
get">get = _getlinecolor(self)
set">set = _setlinecolor(self, value)
```

linewidth

```
get">get = _getlinewidth(self)
set">set = _setlinewidth(self, value)
```

marker

```
get">get = _getmarker(self)
set">set = _setmarker(self, value)
```

markercolor

```
get">get = _getmarkercolor(self)
set">set = _setmarkercolor(self, value)
```

markersize

```
get">get = _getmarkersize(self)
set">set = _setmarkersize(self, value)
```

name

```
get">get = _getname(self)
set">set = _setname(self, value)
```

projection

```
get">get = _getprojection(self)
set">set = _setprojection(self, value)
```

xmtics1

```
get">get = _getxmtics1(self)
set">set = _setxmtics1(self, value)
```

xmtics2

```
get">get = _getxmtics2(self)
set">set = _setxmtics2(self, value)
```

xticlabels1

```
get">get = _getxticlabels1(self)
set">set = _setxticlabels1(self, value)
```

xticlabels2

```
get">get = _getxticlabels2(self)
set">set = _setxticlabels2(self, value)
```

yaxisconvert

```
get">get = _getyaxisconvert(self)
set">set = _setyaxisconvert(self, value)
```

ymtics1

```
get">get = _getymtics1(self)
set">set = _setymtics1(self, value)
```

ymtics2

```
get">get = _getymtics2(self)
set">set = _setymtics2(self, value)
```

yticlabels1

```
get">get = _getyticlabels1(self)
set">set = _setyticlabels1(self, value)
```

yticlabels2

```
get">get = _getyticlabels2(self)
set">set = _setyticlabels2(self, value)
```

Data and other attributes defined here:

```
__slots__ = ['setmember', 'parent', 'name', 'g_name', 'yaxisconvert', 'linecolor', 'line', 'linewidth', 'marker', 'markercolor', 'projection', 'xticlabels1', 'xticlabels2', 'yticlabels1', 'yticlabels2', 'xmtics1', 'xmtics2', 'ymtics1', 'ymtics2']
```

g_name = <member 'g_name' of 'GXy' objects>

parent = <member 'parent' of 'GXy' objects>

setmember = <member 'setmember' of 'GXy' objects>

Functions

```
getGXymember(self, member)
```

```
#####
#
# Function:      getGXymember
#
# Description of Function:
#     Private function that retrieves the Xyvsvy members from the
#     structure and passes it back to Python.
#
#
# Example of Use:
#     return_value =
#     getGXymember(self, name)
#         where: self is the class (e.g., GXy)
#                name is the name of the member that is being
#
#####
```

```
getmember = getGXymember(self, member)
```

```
#####
#
```

```

# Function:      getGXymember
#
# Description of Function:
#     Private function that retrieves the Xyvsvy members from the
#     structure and passes it back to Python.
#
#
# Example of Use:
#     return_value =
#     getGXymember(self, name)
#         where: self is the class (e.g., GXy)
#                name is the name of the member that is being
#
#####

```

renameGXy(self, old_name, new_name)

```

#####
#
# Function:      renameGXy
#
# Description of Function:
#     Private function that renames the name of an existing Xyvsvy
#     graphics method.
#
#
# Example of Use:
#     renameGXy(old_name, new_name)
#         where: old_name is the current name of Xyvsvy graphi
#                new_name is the new name for the Xyvsvy graph
#
#####

```

setGXymember(self, member, value)

```

#####
#
# Function:      setGXymember
#
# Description of Function:
#     Private function to update the VCS canvas plot. If the can
#     set to 0, then this function does nothing.
#
#
# Example of Use:
#     setGXymember(self, name, value)
#         where: self is the class (e.g., GXy)
#                name is the name of the member that is being
#                value is the new value of the member (or att
#
#####

```

setmember = setGXymember(self, member, value)

```
#####  
#  
# Function:      setGXymember  
#  
# Description of Function:  
#     Private function to update the VCS canvas plot. If the can  
#     set to 0, then this function does nothing.  
#  
#  
# Example of Use:  
#     setGXymember(self,name,value)  
#         where: self is the class (e.g., GXy)  
#             name is the name of the member that is being  
#             value is the new value of the member (or att  
#  
#####
```

Data

StringTypes = (<type 'str'>, <type 'unicode'>)